Proceedings of the Peradeniya University Research Sessions, Sri Lanka, Vol.12, Part II, 30th November 2007

Controller Design and Implementation for Cost Effective Mobile Robots

I.B. Wijayasinghe and B.G.L.T. Samaranayake*

Department of Electrical and Electronic Engineering, Faculty of Engineering, University of Peradeniya

Introduction

ROBOCON is an annual robot competition organized by the Asia – Pacific Broadcasting Union (ABU) among the countries in the region. In 2007, the game plan included design and construction of two types of mobile robots, namely manually operated mobile robot, i.e., the manual machine and automatic mobile robot, i.e., the automatic machine. Both the types are supposed to pick hollow cylindrical objects called pearls and put them around a pole called the island. This operation with the manual machine and automatic machines are shown in Figures 1(a) and 1(b) respectively. Depending on the game strategy, there are significant differences between the accessories used in the two types of machines. However, the driving platforms, which need controllers, are identical in terms of the designed acceleration, maximum speed, maneuverability, weight, height of the top most point, ground clearance, etc. Out of them, the main criterions of the design are the speed and the maneuverability as the machines have to travel about 50m in total in the game field in 3 minutes, where the field is of a polygon shape. The speed requirement is over 2 ms^{-1} for a machine of weight of approximately 25 kg with all its accessories connected.

System modeling

Permanent Magnet DC motors are used in the drive platforms, in order to achieve the desired dynamics, because the torque of the stepper motors, which is the other alternative, drastically drops at higher speeds. The block diagram model as shown in Figure 2 is used to model the Permanent Magnet DC motor from the input voltage to the output angular velocity (Figure 3).

In the block diagram, L_a and R_a are the armature inductance and resistance respectively and K is the torque constant, while J being the rotor moment of inertia. Since the manufacturer specified values for the above parameters were not available for the motors, they were identified by measuring armature current I_a and the speed ω for a set of input voltages V_a.

Neglecting the viscose friction, I_a is assumed to be constant at steady state. It can be related to speed as

$$\omega = \left(\frac{1}{K}\right)V - \frac{I_a R_a}{K} \qquad (1)$$

which is an equation of a straight line with gradient 1/K and intercept - I_a R_a/K. The experimental dataset is shown in Table 1 and R_a was calculated for each data set, and the average was taken as the true value of R_a.
The armature current dynamics (Figure 4) corresponding to a voltage step input, with the motor shaft on no load, are used to calculate the parameters L_a, T_s and J.

With *s* being the *Laplace* operator, the voltage and current can be related as

$$I_a(s) = \frac{1}{L_a s + R_a} V(s) \qquad (2)$$

Hence the electrical time constant
$\tau_e = L_a / R_a$ can be calculated from the rise time of the current waveform in Figure 4 and assuming $\tau_e << \tau_m$, where τ_m is the mechanical time constant.

In the current falling, $\tau_m = J R_a / K^2$ is dominant and it can be used to evaluate the relevant parameters. The armature current in steady state is related to the torque T by T=K I_a. The effect of the mass (M) of the mechanical structure can be modeled by

$$T = \frac{2}{Mr^2 s}\omega, \qquad (3)$$

where r is the radius of the road wheels. Hence the resultant inertia felt by the motor can be given by, $J = J_s + J_w + \frac{Mr^2}{2}$. The identified motor parameters are listed in Table 2, which are used in the controller design.

Controller design

The controller mainly consists of speed control and position control. The purpose of the speed controller is to maintain the road wheel speed at the desired value depending on the expected motion profile. Two loops are implemented, one for each motor. But the two loops are coupled as these machines do not have a differential for driving in circular or curved paths and the differential action is implemented electronically through the controllers. Pole placement design technique is used to tune a Proportional Integral (PI) controller ($K_p + K_i/s$), where the gains have been set to obtain a closed loop bandwidth of 10Hz, where the Bode amplitude diagram is shown in Figure 5. An outer position controller is used to ensure the correct position profile is followed. Since the integral action is inherited from the speed loop only a Proportional controller (gain 0.01745) is used here.

Results and conclusions

Speeds of the two road wheels, when they are under different load torques, without the position controller is shown in Figure 5. The corresponding difference of the angle of rotation of the two wheels is shown in Figure 6.

The same scenario, but with the position controller implemented, is shown in Figures 7 and 8 respectively.

It is observed that position controller effectively brings the angular position difference to zero effectively.

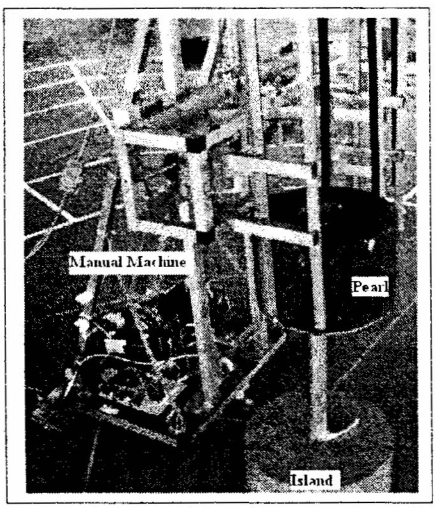

Figure 1 (a): Manual Machine entering a Pearl to the Island

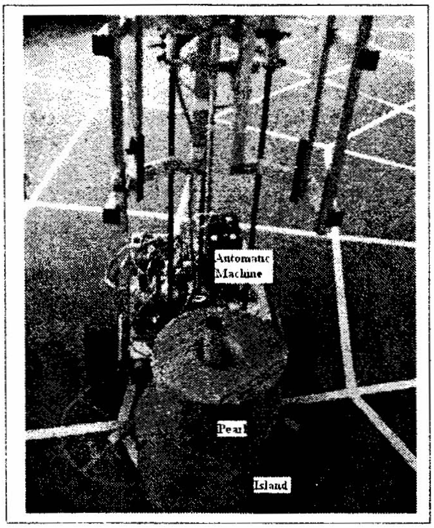

Figure 1 (b): Automatic Machine has just entered a Pearl to the Island

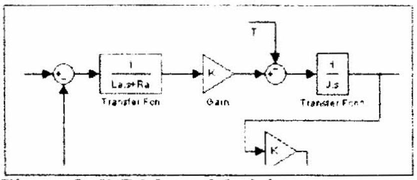

Figure 2: DC Motor Model

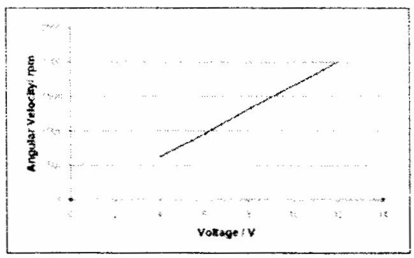

Figure 3: Angular Velocity vs. Voltage

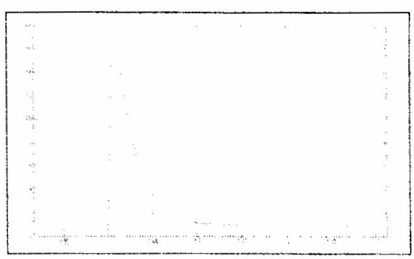

Figure 4: Shape of the Current Trace

Table 1: Steady State Characteristics

Voltage /(V)	Current /(A)	Velocity /(rpm)	Ra /(ohms)
4	0.58	623	0.6742
5	0.61	792	0.6754
6	0.65	958	0.6929
7	0.68	1134	0.6336
8	0.7	1311	0.5793
9	0.73	1482	0.5684
10	0.76	1657	0.5278
11	0.79	1826	0.5344
12	0.81	2004	0.4827

Table 2: Calculated Parameters

Parameter	Value
K	0.05532
Ra	0.5962
La	0.001908
Js	0.0001242
Jw	0.0002967

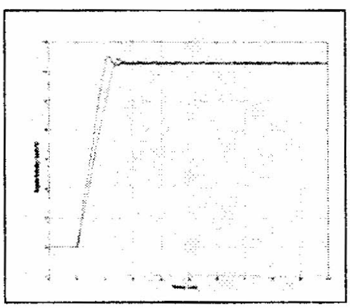

Figure 5: Speeds of the two wheels without Position Controller

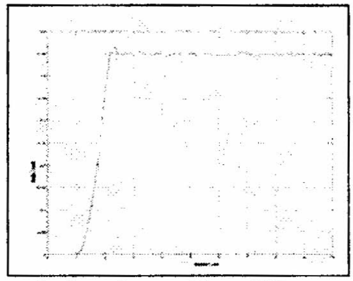

Figure 6: Angular position differences without Position Controller

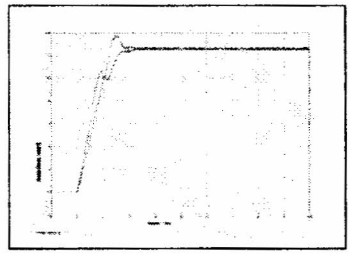

Figure 7: Speeds of the two wheels with the Position Controller

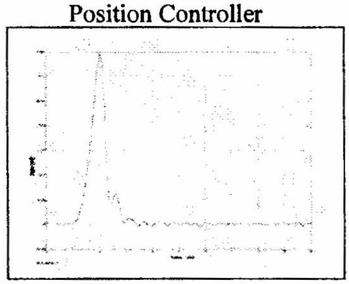

Figure 8: Angular position differences with the Position Controller